

Numerical Solutions to Differential Equations

Lecture Notes #9 — Predictor-Corrector Methods

Peter Blomgren,
 <blomgren.peter@gmail.com>

Department of Mathematics and Statistics
 Dynamical Systems Group
 Computational Sciences Research Center
 San Diego State University
 San Diego, CA 92182-7720

<http://terminus.sdsu.edu/>

Spring 2015

Outline

- 1 Introduction
 - Challenges and Ideas
- 2 Predictor-Corrector Methods
 - Definition and General Ideas
 - Predictor-Corrector Modes
 - Error Analysis, and Estimates
- 3 Predictor-Corrector Methods, ctd.
 - Stability Analysis, Introduction
 - Stability Polynomials
 - Examples: Stability Regions

Implicit Linear Multistep Methods

Suppose we want to solve $y'(t) = f(t, y)$, $y(t_0) = y_0$ by an implicit linear multistep method.

At each step we have to solve the implicit system

$$\mathbf{y}_{n+k} - h\beta_k f(t_{n+k}, \mathbf{y}_{n+k}) = - \sum_{j=0}^{k-1} \alpha_j y_{n+j} + h \sum_{j=0}^{k-1} \beta_j f_{n+j}$$

Usually this is done by the fixed point iteration

$$\mathbf{y}_{n+k}^{[\nu+1]} = h\beta_k f(t_{n+k}, \mathbf{y}_{n+k}^{[\nu]}) - \sum_{j=0}^{k-1} \alpha_j y_{n+j} + h \sum_{j=0}^{k-1} \beta_j f_{n+j}$$

where $y_{n+k}^{[0]}$ is arbitrary (but typically y_{n+k-1}).

Fixed Point Iteration for Implicit LMMs

The fixed point iteration^{Math 541} converges to the unique solution provided that

$$h < \frac{1}{|\beta_k|L},$$

where L is the Lipschitz constant of f with respect to y , i.e.

$$\|f(t, y) - f(t, y + \epsilon)\| \leq L\epsilon, \quad \epsilon > 0.$$

This is usually not very restrictive. In most cases accuracy places tighter constraints on h .

Convergence of Fixed Point Iteration

- Although the fixed point iteration **will converge** for arbitrary starting values $y_{n+k}^{[0]}$, **convergence may be slow** (linear unless we are extremely lucky.)
- Obviously, it would help to have a good initial guess!
- We will obtain the good initial guess from an **explicit** Linear Multistep Method.
- The explicit method is called the **predictor**, and the implicit method the **corrector**. Together they are a **predictor-corrector pair**.

A General Predictor-Corrector Pair

We write a general k -step PC-method:

$$(p) \quad \sum_{j=0}^k \alpha_j^* y_{n+j} = h \sum_{j=0}^{k-1} \beta_j^* f_{n+j}$$

$$(c) \quad \sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j}$$

We will look at different types of predictor-corrector pairs, initially we will be concerned with predictors of Adams-Bashforth type, and correctors of Adams-Moulton type.

Predictor-Corrector Methods

It is an advantage to have the predictor and corrector to be accurate to the same order.

This usually means the step-number for the explicit predictor is greater than that of the implicit corrector, e.g.

$$(p) \quad y_{n+2} - y_{n+1} = \frac{h}{2}(3f_{n+1} - f_n)$$

$$(c) \quad y_{n+2} - y_{n+1} = \frac{h}{2}(f_{n+2} + f_{n+1})$$

is regarded a PC-method with step-number 2, even though the corrector is a 1-step method (and, as written, it also violates $|\alpha_0| + |\beta_0| \neq 0$, i.e. it does not have any term on the n -level).

Predictor-Corrector Modes

Remember:

We are using the predictor to get an initial guess for the fixed point iteration for the corrector method. How many fixed point steps should we take???

[Mode] Correcting to convergence:

In this mode we iterate until

$$\|y_{n+k}^{[\nu+1]} - y_{n+k}^{[\nu]}\| < \epsilon, \quad \text{or} \quad \frac{\|y_{n+k}^{[\nu+1]} - y_{n+k}^{[\nu]}\|}{\|y_{n+k}^{[\nu+1]}\|} < \epsilon,$$

where ϵ usually is of the order of machine-precision (round-off error).

[Mode] Correcting to convergence:

In this mode the predictor plays a very small role. The local truncation error and the linear stability characteristics of the PC-pair are those of the corrector alone.

This mode is not very attractive since we cannot *a priori* predict how many fixed-point iterations will be needed. In a real-time system (e.g. the auto-pilot in an aircraft), this may be dangerous.

At the end of $P(EC)^2$ we have the values $y_{n+k}^{[2]}$ for y_{n+k} and $f_{n+k}^{[1]}$ for $f(t_{n+k}, y_{n+k})$, sometimes we want to update the value of f by performing a further evaluation $f_{n+k}^{[2]} = f(t_{n+k}, y_{n+k}^{[2]})$; this mode would be described as $P(EC)^2E$.

The two classes of modes can be written as

$$P(EC)^\mu E^t, \mu \geq 1, t \in \{0, 1\}.$$

[Mode] Fixed number of Fixed-Point Corrections:

In this mode we perform a fixed number of FP-iteration at each step — usually 1 or 2.

The local truncation error and the linear stability properties of the PC-method depend both on the predictor and corrector (more complicated analysis — more work for us!)

We will use the following short-hand

P	—	Apply the predictor once
E	—	Evaluate f given t and y
C	—	Apply the corrector once

The methods described above are PEC and $P(EC)^2$.

$$P : y_{n+k}^{[0]} = - \sum_{j=0}^{k-1} \alpha_j^* y_{n+j}^{[\mu]} + h \sum_{j=0}^{k-1} \beta_j^* f_{n+j}^{[\mu-1+t]}$$

$$(EC)^\mu : \left\{ \begin{array}{l} f_{n+k}^{[\nu]} = f(t_{n+k}, y_{n+k}^{[\nu]}) \\ y_{n+k}^{[\nu+1]} = - \sum_{j=0}^{k-1} \alpha_j y_{n+j}^{[\mu]} + h \beta_k f_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} \beta_j f_{n+j}^{[\mu-1+t]} \\ \nu = 0, 1, \dots, \mu - 1 \end{array} \right\}$$

$$E^t : f_{n+k}^{[\mu]} = f(t_{n+k}, y_{n+k}^{[\mu]}), \text{ if } t = 1.$$

Error Analysis of $P(EC)^\mu E^t$

[Lambert 105 – 107]

If the predictor is a p^* -order method and the corrector a p -order method, then (using notationally non-consistent LTEs)

$$\begin{aligned} (p) \quad \text{LTE}^*(h) &= C^* h^{p^*+1} y^{(p^*+1)}(\xi^*) + \mathcal{O}(h^{p^*+2}) \\ (c) \quad \text{LTE}(h) &= C h^{p+1} y^{(p+1)}(\xi) + \mathcal{O}(h^{p+2}) \end{aligned}$$

The local truncation error for $P(EC)^\mu E^t$ is $C^{**} h^{p^{**}+1}$, where:

- (i) if $p^* \geq p$ or ($p^* < p$ and $\mu > p - p^*$), $p^{**} = p$ and $C^{**} = C y^{(p+1)}(\xi)$
- (ii) if $p^* < p$ and $\mu = p - p^*$, $p^{**} = p$, but $C^{**} \neq C y^{(p+1)}(\xi)$
- (iii) if $p^* < p$ and $\mu < p - p^*$, $p^{**} = p^* + \mu < p$.

Local Extrapolation

c.f. Richardson Extrapolation. ^{Math 541}

Now that we have an estimate for the error... Why not use that estimate as another correction of the solution?!?

It is really a case of being greedy and trying to eat the cake and still have it. However, local extrapolation (symbol: L) is an accepted feature in many modern codes.

It can be applied in more than one way: $P(ECL)^\mu E^t$, or $P(EC)^\mu LE^t$.

Milne's Error Estimate

If $p^* = p$ it is possible to get an estimate of the leading part of the local truncation error with two subtractions and a multiplication.
 — Something for (almost) nothing!

$$\begin{aligned} (p) \quad \text{LTE}^*(h) &= C^* h^{p+1} y^{(p+1)}(t_n) = y(t_{n+k}) - y_{n+k}^{[0]} + \mathcal{O}(h^{p+2}) \\ (c) \quad \text{LTE}(h) &= C h^{p+1} y^{(p+1)}(t_n) = y(t_{n+k}) - y_{n+k}^{[\mu]} + \mathcal{O}(h^{p+2}) \end{aligned}$$

Subtraction gives

$$(C^* - C) h^{p+1} y^{(p+1)}(t_n) = y_{n+k}^{[\mu]} - y_{n+k}^{[0]} + \mathcal{O}(h^{p+2})$$

Hence (multiply by $\frac{C}{C^* - C}$)

$$\text{LTE}(h) \approx C h^{p+1} y^{(p+1)}(t_n) = \frac{C}{C^* - C} \left[y_{n+k}^{[\mu]} - y_{n+k}^{[0]} \right]$$

$P(EC)^\mu LE^t$

$$P : \quad y_{n+k}^{[0]} = - \sum_{j=0}^{k-1} \alpha_j^* y_{n+j}^{[\mu]} + h \sum_{j=0}^{k-1} \beta_j^* f_{n+j}^{[\mu-1+t]}$$

$$(EC)^\mu : \quad \left\{ \begin{array}{l} f_{n+k}^{[\nu]} = f(t_{n+k}, y_{n+k}^{[\nu]}) \\ y_{n+k}^{[\nu+1]} = - \sum_{j=0}^{k-1} \alpha_j y_{n+j}^{[\mu]} + h \beta_k f_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} \beta_j f_{n+j}^{[\mu-1+t]} \\ \nu = 0, 1, \dots, \mu - 1 \end{array} \right\}$$

$$L : \quad y_{n+k}^{[\mu]} \xleftarrow{\text{update}} \left[1 + \frac{C}{C^* - C} \right] y_{n+k}^{[\mu]} - \left[\frac{C}{C^* - C} \right] y_{n+k}^{[0]}$$

$$E^t : \quad f_{n+k}^{[\mu]} = f(t_{n+k}, y_{n+k}^{[\mu]}), \quad \text{if } t = 1.$$

$P(ECL)^\mu E^t$

$$P : y_{n+k}^{[0]} = - \sum_{j=0}^{k-1} \alpha_j^* y_{n+j}^{[\mu]} + h \sum_{j=0}^{k-1} \beta_j^* f_{n+j}^{[\mu-1+t]}$$

$$(ECL)^\mu : \left\{ \begin{array}{l} f_{n+k}^{[\nu]} = f(t_{n+k}, y_{n+k}^{[\nu]}) \\ y_{n+k}^{[\nu+1]} = - \sum_{j=0}^{k-1} \alpha_j y_{n+j}^{[\mu]} + h \beta_k f_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} \beta_j f_{n+j}^{[\mu-1+t]} \\ \nu = 0, 1, \dots, \mu - 1 \\ y_{n+k}^{[\nu+1]} \xleftarrow{\text{update}} \left[1 + \frac{c}{c^* - c} \right] y_{n+k}^{[\nu+1]} - \left[\frac{c}{c^* - c} \right] y_{n+k}^{[0]} \end{array} \right.$$

$$E^t : f_{n+k}^{[\mu]} = f(t_{n+k}, y_{n+k}^{[\mu]}), \quad \text{if } t = 1.$$

Linear Stability Analysis: Notation

$$\hat{h} = h\lambda$$

$$H = \hat{h}\beta_k$$

$$M_\mu(H) = \frac{H^\mu(1-H)}{1-H^\mu}$$

$$W = \frac{c}{c^* - c}$$

Notice:

$$\lim_{\mu \rightarrow \infty} M_\mu(H) = 0, \quad \text{when } |H| < 1$$

Linear Stability Analysis for Predictor-Corrector Methods

By applying our methods to the linear model problem

$$y'(t) = \lambda y(t), \quad y(t_0) = y_0$$

we can again find the region in $\hat{h} = h\lambda$ space where the method produces non-exponentially growing solutions.

The idea and framework is the same as in our previous cases (LMMs, Runge-Kutta methods), but the algebra involved becomes “somewhat” tedious.

Here, we will summarize some of the key results.

Some Stability Polynomials

$P(EC)^\mu$: (order $2k$ polynomial)

$$\pi(r, \hat{h}) = \beta_k r^k \left[\rho(r) - \hat{h}\sigma(r) \right] + M_\mu(H) \left[\rho^*(r)\sigma(r) - \rho(r)\sigma^*(r) \right]$$

Adding an extra evaluation changes the stability polynomial quite a bit:

$P(EC)^\mu E$: (order k polynomial)

$$\pi(r, \hat{h}) = \rho(r) - \hat{h}\sigma(r) + M_\mu(H) \left[\rho^*(r) - \hat{h}\sigma^*(r) \right]$$

We notice that (in general) the stability polynomials are non-linear in \hat{h} , which means plotting the region of absolute stability \mathcal{R}_A or its boundary, becomes a challenge. [One exception...]

Stability Polynomial in PEC mode

In PEC mode the stability polynomial is linear in \hat{h} :

$$\pi(r, \hat{h}) = \beta_k r^k [\rho(r) - \hat{h}\sigma(r)] + \beta_k \hat{h} [\rho^*(r)\sigma(r) - \rho(r)\sigma^*(r)]$$

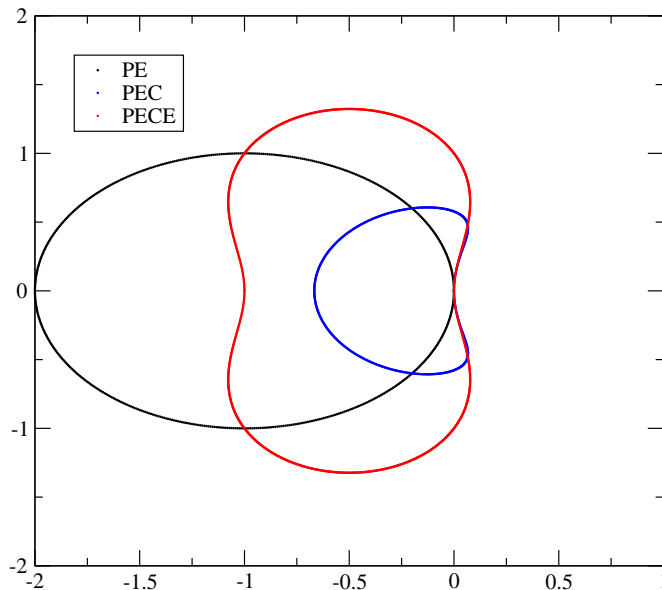
These are easy to plot, but the regions of stability are not great.
 — In fact PEC of order k has a smaller stability region than explicit Adams-Bashforth of the same order!

In general we have to solve a non-linear equation to find the roots of $\pi(r, h)$ — using e.g. Newton's method^{Math 541}.

Adding local extrapolation to the picture makes the stability polynomial more “interesting...”

Stability Regions, PE, PEC, PECE

Order $k = 1$



Stability Polynomials with Local Extrapolation

P(ECL) $^\mu$ E:

$$\pi(r, \hat{h}) = (1+W) [\rho(r) - \hat{h}\sigma(r)] + [M_\mu(H + WH) - W] [\rho^*(r) - \hat{h}\sigma^*(r)]$$

P(ECL) $^\mu$:

$$\pi(r, \hat{h}) = \beta_k r^k \left\{ (1+W) [\rho(r) - \hat{h}\sigma(r)] - W [\rho^*(r) - \hat{h}\sigma^*(r)] \right\} + M_\mu(H + WH) [\rho^*(r)\sigma(r) - \rho(r)\sigma^*(r)]$$

P(EC) $^\mu$ LE:

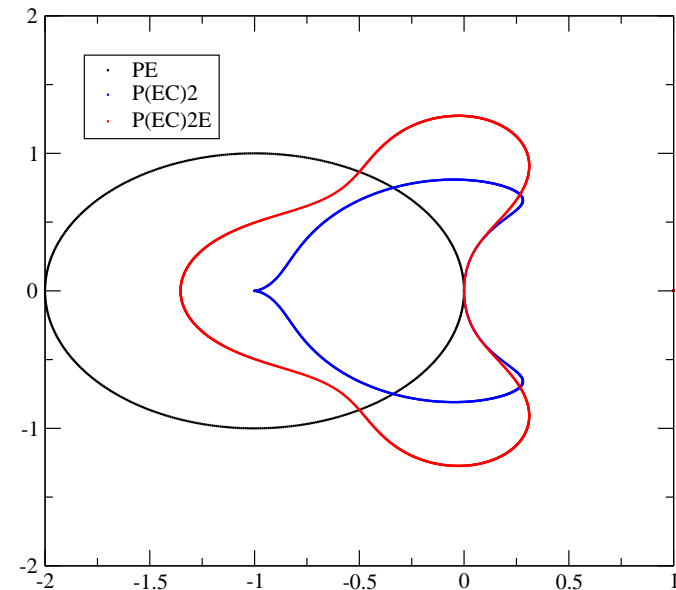
$$\pi(r, \hat{h}) = (1+W) [\rho(r) - \hat{h}\sigma(r)] + [M_\mu(H) + (H-1)W] [\rho^*(r) - \hat{h}\sigma^*(r)]$$

P(EC) $^\mu$ L:

$$\pi(r, \hat{h}) = \beta_k r^k \left\{ (1+W) [\rho(r) - \hat{h}\sigma(r)] - W [\rho^*(r) - \hat{h}\sigma^*(r)] \right\} + [M_\mu(H) + HW] [\rho^*(r)\sigma(r) - \rho(r)\sigma^*(r)]$$

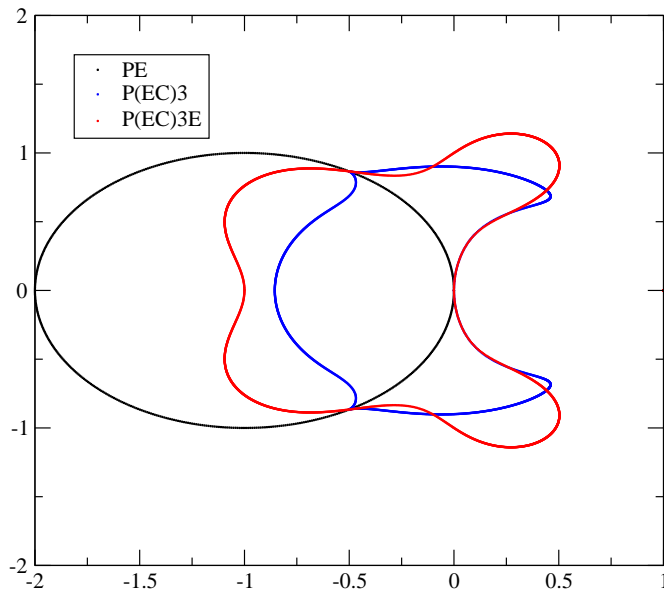
Stability Regions, PE, P(EC) 2 , P(EC) 2 E

$k = 1$



Stability Regions, PE, P(EC)³, P(EC)³E

k = 1



Stability Analysis when k = 1

Predictor, $\rho^*(r) = r - 1, \sigma^*(r) = 1, C^* = 1/2$

$$y_{n+1} - y_n = hf_n$$

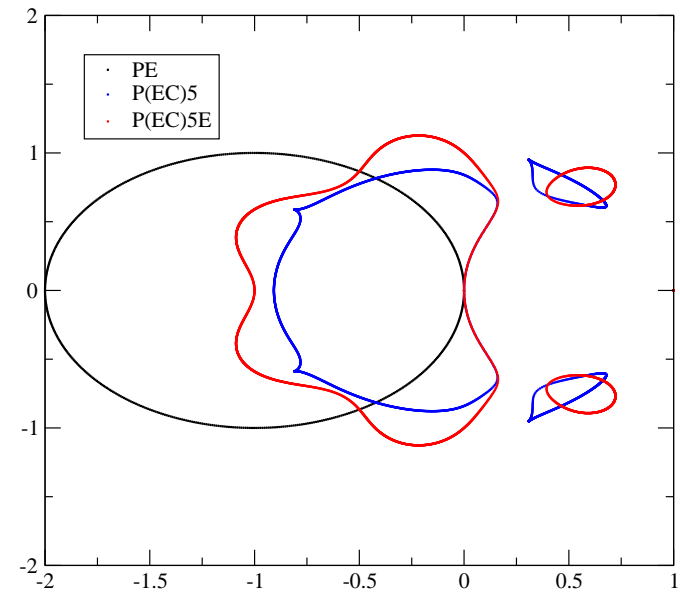
Corrector, $\rho(r) = r - 1, \sigma(r) = r, C = -1/2$

$$y_{n+1} - y_n = hf_{n+1}$$

$$H = h, \quad M_\mu = \frac{h^\mu(1-h)}{1-h^\mu}, \quad W = -\frac{1}{2}$$

Stability Regions, PE, P(EC)⁵, P(EC)⁵E

k = 1



Stability Analysis when k = 1

$P(EC)^\mu$

$$\pi(r, h) = r((r-1) - hr) + \frac{h^\mu(1-h)}{1-h^\mu}((r-1)r - (r-1)1)$$

Multiply through by $1 - h^\mu$ and solve

$$(1 - h^\mu)r((r-1) - hr) + h^\mu(1-h)((r-1)r - (r-1)1) = 0$$

$$h^{\mu+2} [r^2 - (r-1)^2] + h^{\mu+1} [(r-1)^2 - r(r-1)] - hr^2 + r(r-1) = 0$$

Now we can use matlab's friendly roots command to solve for h!

Stability Analysis when $k = 1$ $P(EC)^\mu E$

$$\pi(r, h) = (r - 1) - hr + \frac{h^\mu(1 - h)}{1 - h^\mu} [(r - 1) - h]$$

Multiply through by $1 - h^\mu$ and solve

$$(1 - h^\mu)((r - 1) - hr) + h^\mu(1 - h) [(r - 1) - h] = 0$$
$$h^{\mu+2} - rh + (r - 1) = 0$$

Now we can use matlab's friendly `roots` command to solve for h !

Homework #4, Due 3/20/2015

Pick your favorite Adams-Bashforth (P)redictor (order p^*), and Adams-Moulton (C)orrector (order p) methods, and plot the stability regions for

P(ECL)E

P(ECL)²E

P(EC)LE

P(EC)²LE

Note: The problem is least challenging for $p^* = p = 1 \dots$

Project Idea? — Write a piece of code which can plot the stability regions for any PC-method, as described by $P(ECL^k)^\ell L^m E^n$, ($k + m \leq 1$, $k, m, n \in \{0, 1\}$).